

Using Animations to Learn about Algorithms: An Ethnographic Case Study

Colleen M. Kehoe

John T. Stasko

Graphics, Visualization, and Usability Center

College of Computing

Georgia Institute of Technology

Atlanta, GA 30332-0280

E-mail: colleen & stasko@cc.gatech.edu

Technical Report GIT-GVU-96-20

September 1996

Abstract

A number of studies have found that using animation for explaining dynamic systems had less beneficial effects on learning than hoped. Those results come as a surprise to many instructors and students in computer science where animation is becoming an increasingly popular tool for teaching algorithms. This study takes an *ethnomethodological* approach, observing students using animations and other media to learn about an algorithm. We do not answer the question, “Do animations aid learning?” nor the question “How do they aid learning?” Rather, we gain some insight into how animations can fit into successful learning strategies. Finally, we present several key observations that we hope educators and researchers can use to inform their own work with algorithm animation.

1 Introduction

There is something difficult about understanding and analyzing algorithms; ask any computer science student. What that “something” is and how to reduce the “difficulty” are two problems whose solutions are anxiously awaited by many students and instructors. Meanwhile, guided mainly by intuition, instructors have been looking toward *algorithm animation*[Bro88] as a tool to help their students learn. It is certainly possible to learn about an algorithm without using an animation, but to many it seems almost obvious that students could learn faster and more thoroughly with one. In addition, students using animations reported that they felt the animations assisted them in understanding the algorithm [SBL93]. Imagine the surprise of students and instructors when empirical research about the benefits of animation (in computer science education and elsewhere) began to show disappointing results[RBA90, PE91, SBL93, BCS96].

Rieber, Boyce and Assad conducted a study in 1990[RBA90] using a computer-based science lesson to teach introductory Newtonian mechanics to adults. In short, their results showed that neither the addition of static graphics, nor animated graphics had any effect on learning as measured by a multiple-choice post-test. The study mainly attributes this to a maturation effect: “older students consistently rely less on external images than younger students.[Pre77]” The claim is that adults can and will generate *internal* images given suitable explanations (which the material provided) and therefore the *external* images, the static and animated graphics, were not necessary for learning. On a more promising note, students who viewed animations were able to complete the post-test in significantly less time than the other students. According to the study, the retrieval process requires students to construct images in short-term memory. They hypothesize that the animations aided students in the retrieval process, “presumably by facilitating the initial encoding.”

A study by Palmiter and Elkerton in 1991[PE91] compared the use of animated demonstrations, written text, and a narrated animation for teaching users how to operate a particular graphical interface. They expected, based on the results of earlier studies, that the narrated animation users would perform the best with the animation aiding the initial learning and the narration aiding retention and transfer. Their results showed, however, that the performance of the animation-only and narrated animation groups was very similar; both had problems with retention and transfer. They found evidence that users in the these two groups may have been simply mimicking the procedures and only processing them superficially. As to why the narrations did not have the effect seen with the written text, they give two possible explanations: that auditory text is processed differently from written text, or that users were not paying attention to the narration to process it thoroughly.

Another study was conducted by Stasko, Badre, and Lewis (SBL)[SBL93] in 1993 which used an interactive animation to teach a complicated algorithm to computer science graduate students. Their results showed a “non-significant trend favoring the animation group” in scores for the post-test. The post-test consisted of a set of questions which were each “designed to have one correct answer, and not be open to interpretation.” In particular, the study initially hypothesized that the animation would aid procedural understanding, but the animation group did not perform any better than the control group on questions testing procedural knowledge. They attribute the lack of performance of the animation group to a property of most visualizations—that they represent an expert’s understanding of the algorithm, not a novice’s. “For a student to benefit from the animation, the student must

understand both [the] mapping [from the algorithm to the graphics] and the underlying algorithm on which it is based.... Students just learning about an algorithm do not have a foundation of understanding upon which to construct the visualization mapping. ”

A more recent study conducted by Byrne, Catrambone, and Stasko (BCS) [BCS96] also found limited effects for undergraduates using interactive animations. The study examined the relation of animation to evoking predictions in students. In learning new algorithms, some students viewed animations and some were prompted to make predictions about an algorithm’s operation on novel data sets. For a simple algorithm, the use of animation and/or prediction was beneficial, as measured on a post-test. For a more complex algorithm, however, animation and/or prediction provided no benefit.

Not all studies have had disappointing results, however. In a series of experiments between 1989 and 1992, Mayer *et.al.* demonstrated that illustrations (both static and animated) can have a dramatic positive effect on learning under certain conditions. Results from the early experiments, using only static illustrations, showed that students who viewed labeled illustrations showed better explanative recall and problem-solving transfer than students who saw only labels or illustrations or neither[May89]. Mayer claims that the labeled illustrations played two roles: guiding students attention and helping them build internal connections (i.e. connections between ideas in the text, as opposed to connections to previous knowledge). Another set of experiments in 1991 with Anderson[MA91], considered the use of animations to help students understand scientific explanations. In two experiments, college students without mechanical knowledge viewed animations and/or listened to narrations explaining the operation of a bicycle pump. Students who saw the animation and listened to the narration simultaneously outperformed all other groups on a creative-problem-solving test. They explain this effect with the “*integrated dual-code hypothesis*, adapted from Paivio’s dual-coding theory[Pai71, Pai90, CP91], [which] posits that learners can build both visual and verbal modes of mental representations as well as connections between them.” The claim is that presenting the animation and narration serially does not allow students to build referential connections between the two presentations.

Similarly, Lawrence’s dissertation research showed a positive benefit to the use of animations in after-class laboratory sessions when students were allowed to interact with animations by entering their own data sets as input to algorithms[LBS94].

Gurka and Citrin closely examined experiments such as these in order to better understand the results[GC96]. Their examination has helped to develop a testing model to guide future studies.

1.1 Motivation

All of the studies mentioned either explicitly or implicitly (through their design) test a theory of *how* animations could aid learning. This theory is reflected in the choice of subject matter, the content of the animation, the accompanying materials, the method of presentation, the evaluation of learning, and the tasks and participants chosen. In the studies that have failed to find significant benefits to using animation, at least three explanations seem plausible:

- that there are no or only limited benefits,

- that there are benefits, but the measurements used in the studies are not sensitive to them, or
- that something in the design of the experiment is preventing participants from receiving the benefits, or in other words, the theory of *how* animations could help needs to be re-examined.

This study investigates the third possibility by taking an *ethnomethodological* approach involving detailed observation of students using algorithm animations in educational settings[DMH94]. This approach, which has gained increasing interest recently, was advocated by Hundhausen in a critique of the SBL study[Hun93]. The approach, as he describes it, “relies on the documentation of human situations within the domain of interest as its principle data.” He suggests that the answers to research questions such as “How could animations aid learning?” lie in qualitative data gathered from observing students viewing and interacting with the animations in authentic settings. This is in stark contrast to controlled, comparative psychological studies which usually require settings which are not authentic in order to produce clean, quantitative data. However, we feel that the two approaches can be complementary. In particular, we feel that the results from ethnographic studies can be used to inform the design of and interpret the results of comparative psychological studies.

The design of an experiment to evaluate the use of animation for learning should ideally encourage successful learning strategies, but it must at the very least accommodate them. The purpose of this case study, then, was to gain insight into different ways that animations could fit into successful learning strategies. In particular, we wanted to observe students using animations in a more realistic learning situation to determine:

- what kinds of information the students try to get from the animations and other media
- if the medium chosen provides the information that the student was looking for
- when in the learning process the use of an animation might be helpful.

2 Design of Study

The topic used in the study was the *binomial heap*, a data structure that can be used to implement an abstract data type called a priority queue. Priority queues manage a set of nodes with associated key values and are used in many computer science algorithms. The most basic version of a priority queue involves three operations: *insert*, *extract-minimum*, and *union*. *Insert* simply adds a new node to the priority queue, and *extract-minimum* removes and returns the node with the smallest key value. The *union* operation is performed after each of the others to combine trees of the same size.

Binomial heaps are data structures that can be used to implement priority queues. This data structure and its accompanying algorithms are often taught in advanced undergraduate or graduate level computer science courses. Binomial heaps consist of a forest (ordered set) of binomial trees. Binomial trees are unique in that they always have a size which is a power of two. Binomial trees of equal sizes are combined to make larger trees. The data structure is appealing because all three of its fundamental operations run in logarithmic time. For more details on binomial heaps and their operations, consult any comprehensive computer science algorithms text such as [CLR90].

Three students participated in the observational study, all volunteers and all graduate students at the Georgia Institute of Technology. They had had little or no exposure to binomial heaps, but all had taken undergraduate and graduate level algorithms courses. Because of these qualifications, we considered these students “expert learners” and assumed that they have developed successful strategies for learning new material.

The material on binomial heaps was adapted from a popular algorithms textbook [CLR90] and presented on a page on the World Wide Web. Hotlinks in the text to relevant animations allowed the textual material to be tightly integrated with the animations. When reading about a particular operation, students could simply click on a hotlink to bring up an animation demonstrating the operation. The animations were implemented using the POLKA animation system [SK93]. Three animations were available to students: one illustrated the combination of two small binomial trees into a larger tree and the other two illustrated the *extract-minimum* and *union* operations. Each animation was modeled after a series of original static illustrations from the textbook. The animations contained the same images as the static illustrations, but provided a smooth transition between the images thereby making the relationship between objects in each image more explicit. Binomial heaps involve fairly complex movements of nodes and subtrees, and the animations smoothly illustrated these steps. The animations allowed participants to step forward or backward through the steps of the animation. They could also be displayed simultaneously with the textual material. Figures 1- 4 present still frames from the *extract-min* animation. Each of these frames corresponds to one of the key points of the operation. Note, however, that many in-between transitional frames between consecutive pairs of figures here are not shown.

We wanted to observe students using the materials (text, animations, and still images) to discover what strategies they employed while trying to answer questions about the binomial heap algorithm. Students were given a brief demonstration of the use of the Web browser and POLKA animations. We allowed 10 minutes for students to familiarize themselves with the material before they received a set of questions to answer about the algorithm. The questions covered various aspects of binomial heaps including operations, definitions, mathematical properties, and running times (the questions actually were from the post-test used in the BCS study). Students were then given approximately 35 minutes to answer these questions. Students had full access to the materials while working on the questions; they were not expected to memorize the material. We felt that this arrangement—a “home-work” scenario, rather than an “exam” scenario—was more likely to encourage the kinds of exploration activities we hoped to observe. Providing a specific set of questions to be answered gave direction to the exploration of the materials. Students were encouraged to verbalize their thought processes while viewing the materials and working on the questions. In all sessions, student activity and computer screen activity were video (and audio) taped.

3 Results

We will use “she/her” in the descriptions of the sessions although not all of the students were female. A table summarizing the results follows.

Student 1 had recently received a short (10 minute) introduction to binomial heaps, so she was already somewhat familiar with the operations. She used the animations 3 times during the post-test. The first time was to find out which trees should be combined when

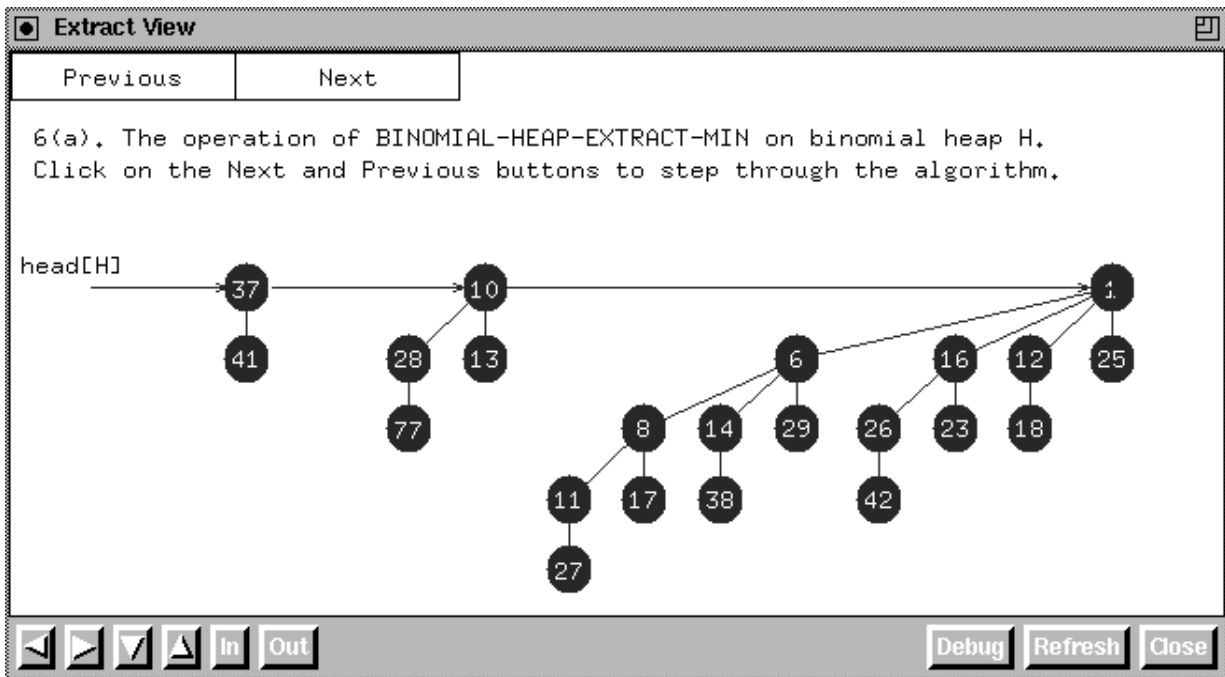


Figure 1: Extract-min animation frame a).

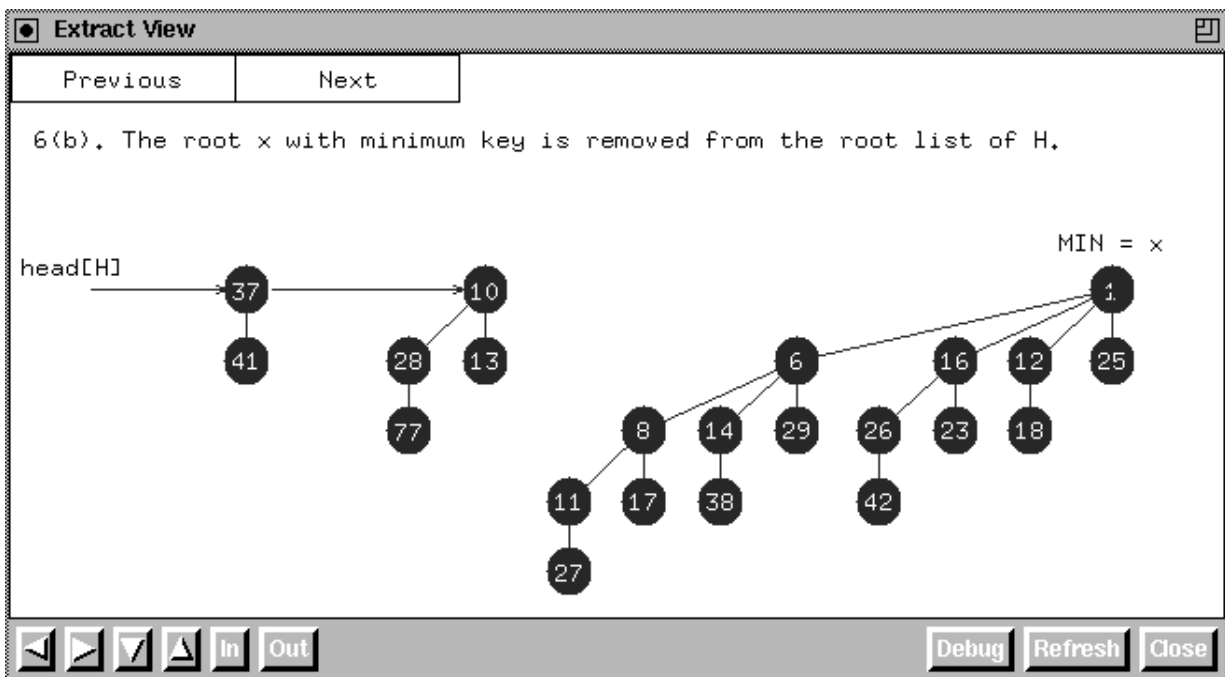


Figure 2: Extract-min animation frame b).

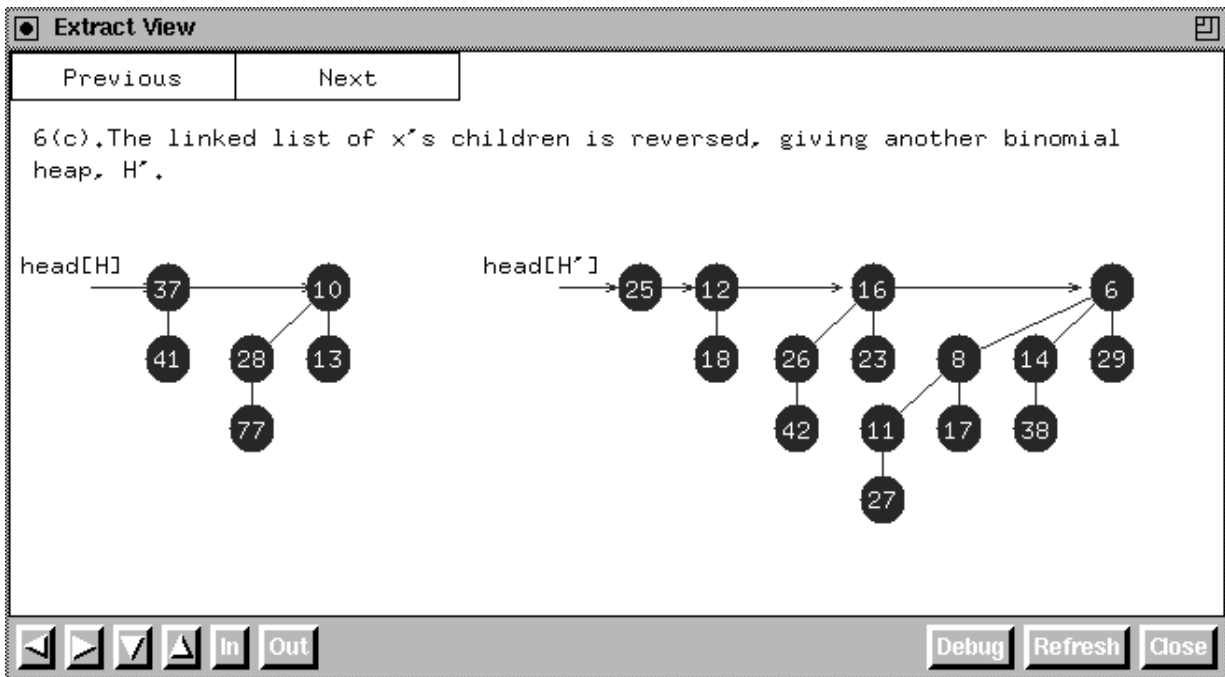


Figure 3: Extract-min animation frame c).

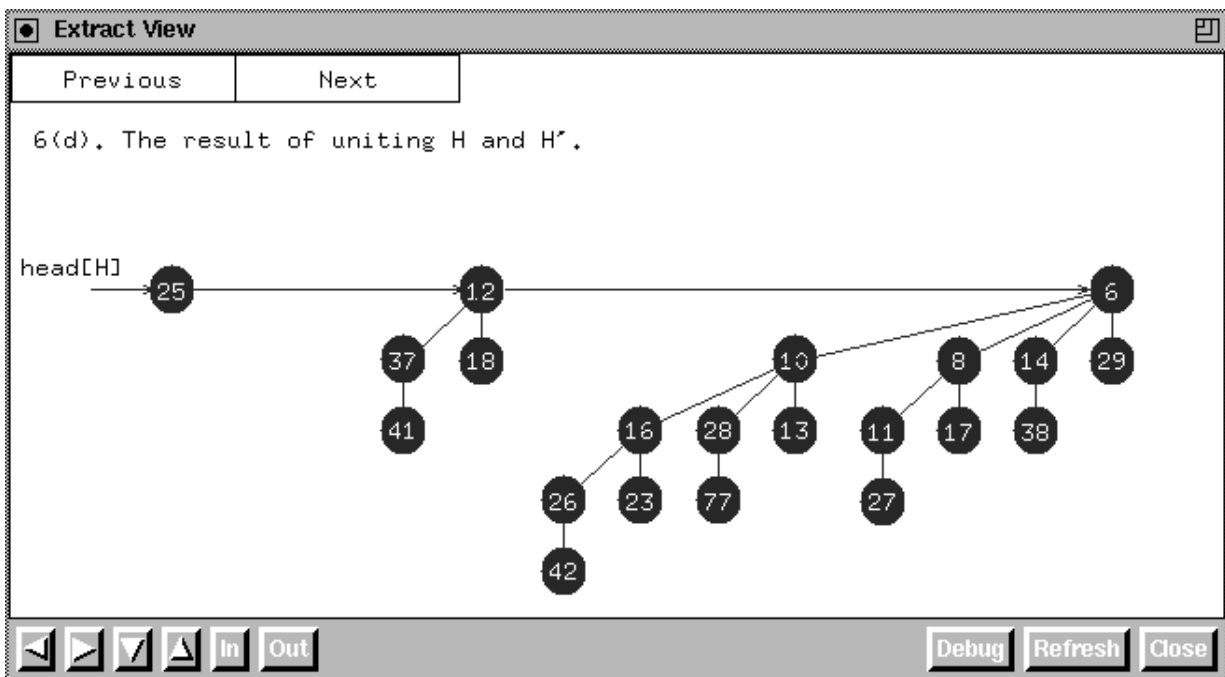


Figure 4: Extract-min animation frame d).

there are 3 trees of the same rank. The other two times were to get the procedure for doing the *extract-minimum* and *union* operations. While watching these animations, the student was reciting the steps to herself: “okay, first you reverse the order of the subtrees, then you line them up...” etc. (paraphrased). She referred back to the text of the page 6 times. Three of those times were to the table of worst-case running times. Once was for terminology (the definition of a B_k tree—a binomial tree of height k). The other two times were to check heap properties (heap ordering and minimum number of binomial trees). She referred to a still image once to determine if the left-most child was “the big one or the small one”. She answered all of the 23 questions correctly.

Student 2 was not familiar with binomial heaps. She had some difficulty operating the animations and the Web browser, but she was able to run at least part of each animation during the first 10 minutes. She also did not realize that she could refer back to the materials until she reached question 12, so she did not use them very much. The first time she wanted to use an animation (*extract-minimum*) she could not locate the link in the page, so she looked at the pseudocode instead. She subsequently was able to locate the *union* animation and run it, but she did not seem to understand the controls, so she saw only the first 2 (of 6) frames of the animation. She went back and forth between the animation and the text several times, but did not seem to find the information she was looking for. She did not use any other materials for the remainder of the exam. A fundamental source of error in her exam was confusion of the term “binomial” with “binary”. She answered 10 of 23 questions correctly.

Student 3 was also unfamiliar with binomial heaps. She used the animations only at the beginning of the post-test but for a substantial amount of time (more than 10 minutes). She went back and forth reading the text and then running both animations. She arranged the windows so she could see the animation and the pseudocode at the same time, making predictions from the pseudocode and testing them with the animation: “now these two should combine and that one should go down here...”. For another question (“build a tree with 13 nodes”), she mentioned that she wanted to run the animation again, but decided she did not have enough time. She referred to the text 8 other times during the post-test. Four of those times were to refer to the table of worst-case running times. Two were for terminology (the definition of a B_k tree and whether the “height” refers to nodes or links). One was to check a heap property (minimum number of binomial trees) and the last one was to check the pseudocode for which trees should be combined when there are 3 trees of the same rank. She referred to the still images 3 times. The first time she was just pointing at a still frame of the animation, working out visually what would happen if a node were inserted. The second time, she was looking for the left-most child, as Student 1 did. The last time she was again looking at a still frame of the animation, working out different cases of *extract-minimum*. She answered 19 of 23 questions correctly.

Table1 presents a summary of the materials used by the three students in the sessions.

3.1 Student Comments

The two students who used the animations the most stated emphatically that the animations helped them understand the algorithm more quickly than the text and still images alone would. One commented that the animations “allowed me to figure out the algorithms without having to read all the pedantic (and hard to understand) pseudocode.” She did

| Information | Animation | Text | Image |
|------------------------|-----------|-------|-------|
| <i>extract-minimum</i> | S1 S2* S3 | S2 S3 | S3 |
| <i>union</i> | S1 S2* S3 | S2 S3 | |
| <i>insert</i> | | | S3 |
| 3 trees, same rank | S1 | S3 | |
| running times | | S1 S3 | |
| heap property | | S1 | |
| minimum # of trees | | S1 S3 | |
| leftmost tree | | | S1 S3 |
| definition of Bk | | S1 S3 | |
| definition of height | | S3* | |

* = student did not obtain the desired information

Table 1: Types of media students (S1, S2, S3) referred to when looking for different types of information.

express some concern, however, that his inferences could be wrong if the animations did not cover all the cases. The other pointed out that some of the text “was essential” to thoroughly understanding the algorithm, but that the animations helped her “create a visualization.” The third student said that she did derive some benefit from the animations. He felt that more examples and better explanations were needed before the animations would be really valuable.

4 Discussion

In this section we utilize observations from this study to help answer questions posed earlier.

What kinds of information do students try to get from the animations and other media? As we would expect, all three students turned to the animation when they wanted to understand the series of steps in the operation (either *extract-minimum* or *union*). Interestingly, two of the three students also made use of the textual description (i.e. pseudocode) while trying to learn the steps of the operations. They went back and forth between the text and animation several times, perhaps trying to build connections between the two representations. One of the students was verbally making predictions based on the pseudocode and then testing them with the animation. The one student who did not use the text in conjunction with the animations was already somewhat familiar with the operations. Also, this student used the animations on one other occasion: to find out what happened in a particular case of *extract-minimum* (three trees of the same rank). She later commented that she ran the animation at this point because she assumed it would cover all the important cases. Another student encountered the same case, recalled seeing that case handled in the pseudocode and turned there instead. Students did not make use of the first animation demonstrating the combination of two binomial heaps except to view it once during the first 10 minutes. This is probably due to the fact that it is a very simple operation compared to the others.

The text was the most heavily used medium by all three students. As mentioned above,

the pseudocode was used extensively by two of the students for understanding the steps of the operations. The text also contained a table of worst-case running times of the operations which two of the students used directly to answer the questions. The third produced the running times from memory. Students also used the text to clarify definitions, terminology, and invariant properties of the data structure.

The least frequently used medium was the still images, including still frames of the animation. (They might have played a larger role, however, if the animations were not available.) Two of the three students referred to a still image when trying to decide if the leftmost child of node was “the big one.” This is probably because the designation “leftmost” makes more sense in an image than in a textual description. In this case, the animation was not needed because the question referred to a static object (a tree of size 128) rather than an operation on an object. Student 3 made a very interesting use of the still images. She often used them as a starting point for performing mental operations—testing what the result of various operations would “look” on the particular heap.

Does the medium chosen provide the information the student was looking for? In almost every case, the students found the information they were looking for in the chosen medium. Students seemed confident about the answers they gave after looking at the materials and were able to move through the questions fairly quickly. After only 10 minutes of scanning the material, the students seemed to have developed a good sense of what information was contained in the different media and perhaps had developed an index of some sort. Several times students remarked that they had “read about X” or “saw an animation about Y” even if they did not recall the specific details.

One notable exception was Student 2, who had trouble with the interface of the Web browser and the POLKA animation system. When she did attempt to use the animations, she was not able to find the information she was looking for because she was only seeing the first two of six frames. She became frustrated with the system and gave up on both the animations and the browser. She worked from memory for the majority of the questions. This student’s experience emphasizes an important point: that additional technology can be as much of a hindrance as a help depending on the student.

When in the learning process might the use of an animation be helpful? The two students who used the animations extensively showed very different patterns of use. Student 1, who was already somewhat familiar with the operations, used the animations when she had a specific need to refine her knowledge of a particular operation. When she encountered a question that required a more detailed understanding of an operation, she ran the related animation until she understood the steps enough to solve the problem. She referenced the animations, text and still images throughout the session. Student 3, who had no previous experience with the operations, took a different approach. After glancing over the first few questions she realized that she did not have a detailed enough understanding of the algorithm to answer them. She spent the next 12 minutes systematically going through each animation and simultaneously tracing through the pseudocode. When she was satisfied that she had a sufficient understanding of the operations, she returned to the questions. She did reference the text and still images later in the session, but she did not run the animations again.

The animations seemed to play several distinct roles in the learning processes of the students. Both students used the animations to get an overall impression of the effect of the operation on the data structure. Student 1 then used them as a way to refine to her

understanding of the operations with more detail. Student 3 used the animations to test her understanding of operations by making and testing predictions.

5 Conclusion

Based on these observations, we have identified several key points that we hope researchers and educators will consider in their future use of instructive animations:

Students use animations showing the steps of an operation simply for learning the steps of the operation.

They did not seem to make any inferences about the general properties of the algorithm such as running times, mathematical properties, or algorithm invariants. This could be because there were not enough examples from which they could generalize. Alternatively, it could simply be the case that it was easier to read the text about the general properties of the algorithm than to infer them from the animation.

Students report that animations help them learn the operations faster than they would otherwise.

The students claim that they are able to learn faster because they do not have to spend time developing their own mental visualizations of the operations. There is a danger here in relying on students' reports, however; it is not clear whether students can accurately evaluate their own learning processes.

Students exhibited a variety of successful learning strategies in which animations seemed to be useful.

It may not be possible to measure the utility of an animation "in general." Instead, researchers and educators should consider trying to identify the particular learning strategies in which they can be useful.

Students need to build connections between the representations (animation, text, pseudocode) for them to be useful.

Some students can spontaneously build these connections, but this process should be explicitly supported in the design of the system for the maximum benefit. The first step toward this is to allow the simultaneous display of the different representations.

Clearly, researchers and instructors are interested in the fundamental question of whether animations can assist students learn and understand algorithms. Formal, comparative experiments seeking to answer that question must decide how the animations will be used by the students. Although case studies such as this one cannot answer the fundamental question above, they can provide experimenters with data and experience about how students interact with animations and about the learning strategies utilized by students.

6 Acknowledgments

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship. Thanks to Mark Guzdial, Richard Catrambone, and Cindy Hmelo for their comments and suggestions. Special thanks to the students who now know more about binomial heaps than they ever wanted to know.

References

- [BCS96] Michael D. Byrne, Richard Catrambone, and John T. Stasko. Do algorithm animations aid learning? Technical Report GIT-GVU-96-18, Graphics, Visualization, and Usability Center, Georgia Institute of Technology, Atlanta, GA, August 1996.
- [Bro88] Marc H. Brown. Perspectives on algorithm animation. In *Proceedings of the ACM SIGCHI '88 Conference on Human Factors in Computing Systems*, pages 33–38, Washington D.C., May 1988.
- [CLR90] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 1990.
- [CP91] J. M. Clark and A. Paivio. Dual coding theory and education. *Educational Psychology Review*, 3:149–210, 1991.
- [DMH94] Sarah Douglas, Donna McKeown, and Christopher Hundhausen. Exploring human visualization of algorithms. Technical Report CIS-TR-94-27, University of Oregon, Eugene, OR, 1994.
- [GC96] Judith S. Gurka and Wayne Citrin. Testing effectiveness of algorithm animation. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, pages 182–189, Boulder, CO, September 1996.
- [Hun93] Christopher D. Hundhausen. Subverting the comparative research paradigm: The potential for ethnomethodology in evaluating the effects of algorithm visualization on learning. Unpublished seminar paper, Department of Computer and Information Science, University of Oregon, 1993.
- [LBS94] Andrea W. Lawrence, Albert M. Badre, and John T. Stasko. Empirically evaluating the use of animations to teach algorithms. In *Proceedings of the 1994 IEEE Symposium on Visual Languages*, pages 48–54, St. Louis, October 1994.
- [MA91] Richard E. Mayer and Richard B. Anderson. Animations need narrations: An experimental test of a dual-coding hypothesis. *Journal of Educational Psychology*, 83(4):484–490, 1991.
- [May89] Richard E. Mayer. Systematic thinking fostered by illustrations in scientific text. *Journal of Educational Psychology*, 81(2):240–246, 1989.
- [Pai71] A. Paivio. *Imagery and Cognitive Processes*. Holt, Rinehart and Winston, New York, 1971.
- [Pai90] A. Paivio. *Mental Representations: A Dual Coding Approach*. Oxford University Press, New York, 1990.
- [PE91] Susan Palmiter and Jay Elkerton. An evaluation of animated demonstrations for learning computer-based tasks. In *Proceedings of the ACM SIGCHI '91 Conference on Human Factors in Computing Systems*, pages 257–263, New Orleans, LA, May 1991.

- [Pre77] M. Pressley. Imagery and children's learning: Putting the picture in developmental perspective. *Review of Educational Research*, 47(4):585–622, 1977.
- [RBA90] Lloyd P. Rieber, Mary J. Boyce, and Chahriar Assad. The effects of computer animation on adult learning and retrieval tasks. *Journal of Computer-Based Instruction*, 17(2):46–52, Spring 1990.
- [SBL93] John Stasko, Albert Badre, and Clayton Lewis. Do algorithm animations assist learning? An empirical study and analysis. In *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems*, pages 61–66, Amsterdam, Netherlands, April 1993.
- [SK93] John T. Stasko and Eileen Kraemer. A methodology for building application-specific visualizations of parallel programs. *Journal of Parallel and Distributed Computing*, 18(2):258–264, June 1993.